IT223 (ADVANCE DATABASE SYSTEM)
MIDTERM PROJECT – SYSTEM DOCUMENTATION

Proponents: **Balansag, John Albert Q.**
Canoy, Rommel A.
Vasquez, Regie T.

Course Year-set: BSIT-2A
Date Submitted: 3/14/2020

TITLE
OBS-Online Banking System

BUSINESS PROCESS

Banking Activities are considered to be the life blood of the economy. The banking industry is witnessing a revolution in products, process, markets and regulations and a revolution that is not about to stop or even slow down. Since the only option is to adapt and evolve, it is essential that systems have the flexibility to quickly adjust the need of today's financial market. It is a tough challenge, because today's fast-moving marketplace is also extremely competitive. Moreover, the need to retain existing customers and attract new ones often conflicts with the need to reduce costs and improve efficiency. But whatever the challenges facing in retail banking operation, Online Banking System - OBS can help to meet and overcome them.

Withdrawing and depositing with traditional banks can be a drag or can be a tedious thing to do. But with online banking, consumers aren't required to visit a bank branch to complete most of their basic banking transactions. They can do all of this at their own convenience, wherever they want—at home, at work, or on the go.

The system is an online based application and can support online registration. Each user can own one credit card number and can utilize the loan management function, the user can also withdraw and deposit money, view transaction history and transfer money to another account. Users have classification which we called super user, user, and staff. Super user can give permissions, group, lock, freeze users account, it can also change and view user's sensitive information and can also add staff accounts. Staff on the other hand, can only send notifications to the users and view the user's transaction history.

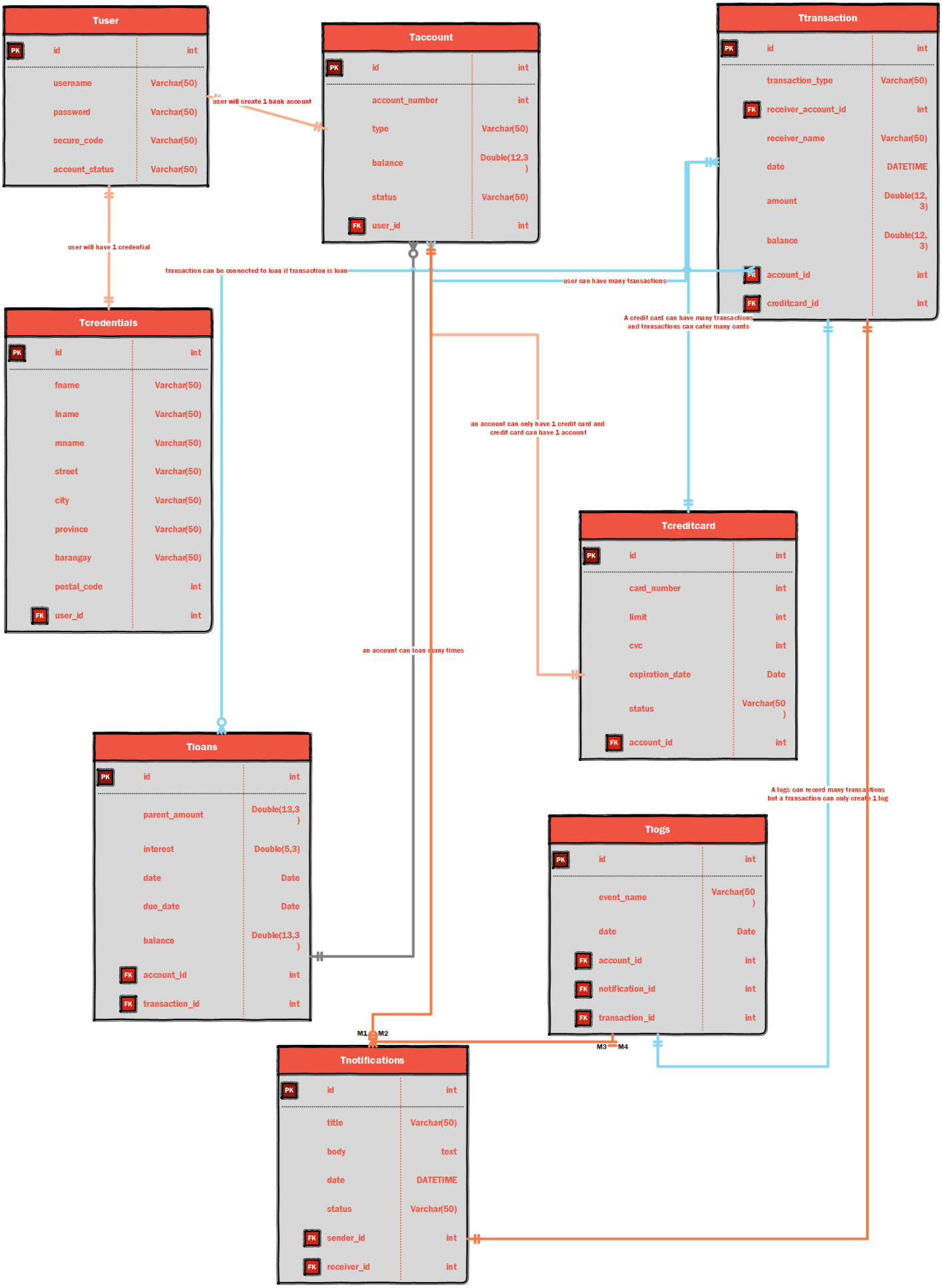
This system also utilizes charts and graphs so that users can manage and monitor its expenses or spending. This system can also be scaled so the developers can add more functions without affecting its main functionality.

SYSTEM FUNCTIONS

1. **Registration of system user:**
    a. Customer Account
    b. Admin Account
2. **System module:**
    a. Customer Creation Form.
    b. Existing customer details
    c. Customer Access Form
    d. Each customer login identified by Access Code and Account No.
    e. Credit Card Management
    f. Loan Management
    g. Balance Inquiry
    h. System Logs
3. **Admin module:**
    a. Add Employees

    b. Group Users
    c. Give permissions
    d. Lock User Account
    e. Freeze User Account
4. **Banking Main Menu:**
    a. Transaction – Debit, Credit, Withdraw, Send
    b. User Information Details
    c. User Statistics
5. Transaction Summary, Reports
6. Account Closing
7. Notification System

ERD (ENTITY-RELATIONSHIP DIAGRAM)


Tuser
PK id int
username Varchar(50)
password Varchar(50)
secure_code Varchar(50)
account_status Varchar(50)
user will create 1 bank account
Taccount
PK id int
account_number int
type Varchar(50)
balance Double(12,3)
status Varchar(50)
FK user_id int
Ttransaction
PK id int
transaction_type Varchar(50)
FK receiver_account_id int
receiver_name Varchar(50)
date DATETIME
amount Double(12,3)
balance Double(12,3)
FK account_id int
FK creditcard_id int
user will have 1 credential
transaction can be connected to loan if transaction is loan
user can have many transactions
A credit card can have many transactions and transactions can cater many cards
Tcredentials
PK id int
fname Varchar(50)
lname Varchar(50)
mname Varchar(50)
street Varchar(50)
city Varchar(50)
province Varchar(50)
barangay Varchar(50)
postal_code int
FK user_id int
an account can only have 1 credit card and credit card can have 1 account
Tcreditcard
PK id int
card_number int
limit int
cvc int
expiration_date Date
status Varchar(50)
FK account_id int
an account can loan many times
Tloans
PK id int
parent_amount Double(13,3)
interest Double(5,3)
date Date
due_date Date
balance Double(13,3)
FK account_id int
FK transaction_id int
A logs can record many transactions but a transaction can only create 1 log
Tlogs
PK id int
event_name Varchar(50)
date Date
FK account_id int
FK notification_id int
FK transaction_id int
M1 M2
M3 M4
Tnotifications
PK id int
title Varchar(50)
body text
date DATETIME
status Varchar(50)
FK sender_id int
FK receiver_id int

DATA DICTIONARY

tUser

FIELD NAME	DATATYPE	KEY TYPE	DESCRIPTION
id	INTEGER	PRIMARY KEY	Id for the user.
username	VARCHAR(50)	None	Username for the user account.
password	VARCHAR(50)	None	Password for the user account.
secure_code	VARCHAR(50)	None	User unique identification code.
account_status	VARCHAR(50)	None	Status of the account.

tCredentials

FIELD NAME	DATATYPE	KEY TYPE	DESCRIPTION
id	INTEGER	PRIMARY KEY	Id for credentials.
fname	VARCHAR(50)	None	First name of the user.
lname	VARCHAR(50)	None	Last name of the user.
mname	VARCHAR(50)	None	Middle name of the user.
street	VARCHAR(50)	None	Street address of the user.
city	VARCHAR(50)	None	City address of the user.
province	VARCHAR(50)	None	Province address of the user.
barangay	VARCHAR(50)	None	Barangay address of the user.
postal_code	INTEGER	None	Postal code address of the user.
user_id	INTEGER	FOREIGN KEY	Id from the user table.

tAccount

FIELD NAME	DATATYPE	KEY TYPE	DESCRIPTION
id	INTEGER	PRIMARY KEY	Id for the account.
account_number	INTEGER	None	Account_number for the account.
type	VARCHAR(50)	None	Type of the account.
balance	DOUBLE(12,3)	None	Balance of the account.
status	VARCHAR(50)	None	Status of the account.
user_id	INT	FOREIGN KEY	User id from User table.

tTransaction

FIELD NAME	DATATYPE	KEY TYPE	DESCRIPTION
id	INTEGER	PRIMARY KEY	Id for the transaction.
transaction_type	VARCHAR(50)	None	Type of transaction.

receiver_account_id	INTEGER	FOREIGN KEY	Receiver ID if the transaction has been sent.
receiver_name	VARCHAR(50)	None	Name of the receiver.
date	DATETIME	None	Date and time of the transaction.
amount	DOUBLE(12,3)	None	Amount money in the transaction.
balance	DOUBLE(12,3)	None	New balance after transaction
account_id	INTEGER	FOREIGN KEY	Id of the account table
creditcard_id	INTEGER	FOREIGN KEY	Id from credit card table

tCredit card

FIELD NAME	DATATYPE	KEY TYPE	DESCRIPTION
id	INTEGER	PRIMARY KEY	Id for the credit card.
card_number	INTEGER	NONE	Card number for the credit card
limit	INTEGER	None	
cvc	INTEGER	None	
expiration_date	DATE	None	Date for expiration
status	VARCHAR(50)	None	Status of the credit card
account_id	INTEGER	FOREIGN KEY	Id from the account table

tLoans

FIELD NAME	DATATYPE	KEY TYPE	DESCRIPTION
id	INTEGER	PRIMARY KEY	Id for the loan.
parent_amount	DOUBLE(13,3)	NONE	Original amount loaned.
interest	DOUBLE(5,3)	None	Interest for the amount loaned.
date	DATE	None	Date loaned.
due_date	DATE	None	Due date.
balance	DOUBLE(13,3)	None	Loaned balance.
account_id	INTEGER	FOREIGN KEY	Id from the account table
transaction_id	INTEGER	FOREIGN KEY	Id from the transaction table

tLogs

FIELD NAME	DATATYPE	KEY TYPE	DESCRIPTION
id	INTEGER	PRIMARY KEY	Id for the logs.
event_name	VARCHAR(50)	None	Event triggered.
date	DATETIME	None	Date and time of the event triggered.
account_id	INTEGER	FOREIGN KEY	Id from the account table
notification_id	INTEGER	FOREIGN KEY	Id from notifications table
transaction_id	INTEGER	FOREIGN KEY	Id from the transaction table.

tNotifications

FIELD NAME	DATATYPE	KEY TYPE	DESCRIPTION
id	INTEGER	PRIMARY KEY	Id for the notifications.
title	VARCHAR(50)	None	Title of the notification.
body	TEXT	None	Body message of the notification.
date	DATETIME	None	Date of the notification.
status	VARCHAR(50)	None	Read or unread.
sender_id	INTEGER	FOREIGN KEY	Id from user(sender) table
receiver_id	INTEGER	FOREIGN KEY	Id from user(receiver) table